

1 **CLAIMS**

2

3 1. A method of synchronizing objects between first and second object
4 stores, comprising:

5 maintaining a reference store containing identifying data segments
6 corresponding respectively to a plurality of objects that are to be synchronized
7 between the first and second object stores;

8 submitting a particular identifying data segment from the reference store to
9 a first interface associated with an application program that maintains the first
10 object store; and

11 querying a second interface associated with the application program to
12 determine whether the object corresponding to said particular identifying data
13 segment has been changed in the first object store.

14

15 2. One or more computer-readable media comprising computer
16 executable instructions for performing the method recited in claim 1.

17

18 3. A method as recited in claim 1, further comprising updating the
19 second object store with the changed object from the first object store if the object
20 has been changed in the first object store.

1 **4.** A method as recited in claim 1, further comprising:
2 determining whether the object corresponding to said particular identifying
3 data segment has been changed in the second object store; and
4 updating the first object store with the changed object from the second
5 object store if the object has been changed in the second object store.

6
7 **5.** A method as recited in claim 1, further comprising receiving the
8 identifying data segments from one or more interfaces associated with the
9 application program, wherein each identifying data segment contains an object
10 identifier for a corresponding object in the first object store.

11
12 **6.** A method as recited in claim 1, further comprising receiving the
13 identifying data segments from one or more interfaces associated with the
14 application program, wherein some of the identifying data segments contain data
15 representing properties that are supported by the second object store but that are
16 not supported by the first object store.

1 7. A method of synchronizing objects between first and second object
2 stores, comprising:

3 maintaining a reference store containing identifying data segments
4 corresponding respectively to a plurality of objects that have previously been
5 synchronized between the first and second object stores;

6 submitting a particular identifying data segment from the reference store to
7 a first interface method associated with an application program that maintains the
8 first object store;

9 in response to submitting said particular identifying data segment from the
10 reference store, receiving a handle that corresponds to said particular identifying
11 data segment; and

12 querying a second interface associated with the application program with
13 the received handle to determine if the object corresponding to said particular
14 identifying data segment has been changed in the first object store since the last
15 synchronization.

16
17 8. One or more computer-readable media comprising computer
18 executable instructions for performing the method recited in claim 7.

19
20 9. A method as recited in claim 7, further comprising if the object has
21 been changed in the first object store, updating the second object store with the
22 changed object from the first object store.

1 **10.** A method as recited in claim 7, further comprising:

2 maintaining a mapping between identifying data segments and object

3 identifiers that are used in the second store to identify objects; and

4 submitting the object identifier that is mapped to the particular identifying
5 data segment to a program that maintains the second object store to determine
6 whether the object corresponding to said particular identifying data segment has
7 been changed in the second object store.

8

9 **11.** A method as recited in claim 7, further comprising:

10 maintaining a mapping between identifying data segments and object
11 identifiers that are used in the second store to identify objects;

12 submitting the object identifier that is mapped to the particular identifying
13 data segment to a program that maintains the second object store to determine
14 whether the object corresponding to said particular identifying data segment has
15 been changed in the second object store; and

16 if the object has been changed in the second object store, updating the first
17 object store with the changed object from the second object store.

18

19 **12.** A method as recited in claim 7, further comprising receiving the
20 identifying data segments from one or more interfaces associated with the
21 application program, wherein each identifying data segment contains an object
22 identifier for a corresponding object in the object store.

1 **13.** A method as recited in claim 7, further comprising receiving the
2 identifying data segments from one or more interfaces associated with the
3 application program, wherein some of the identifying data segments contain data
4 representing properties that are supported by the second object store but that are
5 not supported by the first object store.
6

7 **14.** One or more computer-readable media comprising computer
8 executable instructions that, when executed, direct a computing system to
9 synchronize objects between first and second object stores, the instructions
10 comprising:
11

12 maintaining a reference store containing identifying data segments
13 corresponding respectively to a first group of objects;
14

15 submitting the identifying data segments to a first interface associated with
16 an application program that maintains the first object store, wherein the first object
17 store includes a second group of objects that potentially includes at least some of
18 the first group of objects;
19

20 in response to submitting said identifying data segments, receiving a first
21 list of handles, said handles corresponding respectively to the objects of the first
22 group;
23

24 querying a second interface associated with the application program for a
25 second list of handles, said handles corresponding respectively to the objects of
the second group;
26

27 matching any handles of the first and second lists that correspond to the
28 same object; and
29

1 querying a third interface associated with the application program with
2 matching handles from the first and second lists to determine if the object to which
3 they correspond has been changed in the first object store.

4

5 **15.** One or more computer-readable media as recited in claim 14,
6 wherein said matching comprises querying a fourth interface associated with the
7 application program to determine whether handles of the first and second lists
8 correspond to the same object.

9

10 **16.** One or more computer-readable media as recited in claim 14, further
11 comprising computer executable instructions that, when executed, perform
12 concluding that a particular object has been deleted from the first object store if
13 the handle from the first list corresponding to said particular object does not have a
14 matching handle from the second list.

15

16 **17.** One or more computer-readable media as recited in claim 14, further
17 comprising computer executable instructions that, when executed, perform
18 concluding that a particular object has been added to the first object store if the
19 handle from the second list corresponding said particular object does not have a
20 matching handle from the first list.

21

22 **18.** One or more computer-readable media as recited in claim 14, further
23 comprising computer executable instructions that, when executed, perform
24 updating the second object store with any objects that have been changed in the
25 first object store.

1
2 **19.** One or more computer-readable media as recited in claim 14, further
3 comprising computer executable instructions that, when executed, perform:

4 receiving a modified form of a particular object from the second object
5 store;

6 identifying which handle from the second list corresponds to said particular
7 object;

8 providing the identified handle and the modified form of said particular
9 object to a fourth interface associated with the application program; and

10 instructing the application program to replace the object corresponding to
11 the identified handle with the modified form of the object.

12
13 **20.** One or more computer-readable media as recited in claim 14, further
14 comprising computer executable instructions that, when executed, perform:

15 maintaining a mapping between handles and object identifiers that are used
16 in the second store to identify objects;

17 receiving a modified form of a particular object from the second object
18 store;

19 receiving an object identifier corresponding to said particular object;

20 referencing the mapping with the object identifier to identify which handle
21 corresponds to said particular object;

22 providing the identified handle and the modified form of said particular
23 object to a fourth interface associated with the application program; and

24 instructing the application program to replace the object corresponding to
25 the identified handle with the modified form of the object.

1
2 **21.** One or more computer-readable media as recited in claim 14, further
3 comprising computer executable instructions that, when executed, perform:

4 receiving a modified form of a particular object from the second object
5 store;

6 identifying which handle from the second list corresponds to said particular
7 object;

8 providing the identified handle and the modified form of said particular
9 object to a fourth interface associated with the application program;

10 instructing the application program to replace the object corresponding to
11 the identified handle with the modified form of the particular object;

12 requesting new identifying data from a fifth interface associated with the
13 application program for the object corresponding to the identified handle; and

14 updating the reference store with the new identifying data.

15
16 **22.** One or more computer-readable media as recited in claim 14, further
17 comprising computer executable instructions that, when executed, perform:

18 requesting new identifying data for a particular object from a fourth
19 interface associated with the application program if the particular object has been
20 changed in the first object store; and

21 updating the reference store with the new identifying data for said particular
22 object.

1 **23.** One or more computer-readable media as recited in claim 14, further
2 comprising computer executable instructions that, when executed, perform
3 maintaining a mapping between handles and object identifiers that are used in the
4 second store to identify objects.

5
6 **24.** One or more computer-readable media as recited in claim 14, further
7 comprising computer executable instructions that, when executed, perform:

8 maintaining a mapping between handles and object identifiers that are used
9 in the second store to identify objects; and

10 maintaining a mapping between the identifying data segments and the
11 object identifiers.

12
13 **25.** One or more computer-readable media as recited in claim 14, further
14 comprising computer executable instructions that, when executed when a
15 particular object has been changed in the first object store, perform:

16 receiving a modified form of the particular object from the second object
17 store;

18 identifying which handle from the second list corresponds to said particular
19 object;

20 providing the identified handle and the modified form of said particular
21 object to a fourth interface associated with the application program;

22 querying a fifth interface associated with the application program for
23 conflict resolution text; and

24 prompting a user using the conflict resolution text.

1 **26.** A set of application program interfaces embodied on a computer-
2 readable medium for execution on a computer in conjunction with an application
3 program that maintains an object store, comprising:

4 a first interface that receives an identifying data segment and that returns a
5 handle corresponding to the identifying data segment; and

6 a second interface that receives the handle and in response returns an
7 indication of whether an object corresponding to the identifying data segment has
8 been changed in the object store.

9
10 **27.** A set of application program interfaces as recited in claim 26,
11 wherein the second interface also receives a second handle, wherein the second
12 interface compares identifying data segments corresponding to the received
13 handles to determine whether the object has been changed in the object store.

14
15 **28.** A set of application program interfaces as recited in claim 26,
16 further comprising a third interface that enumerates a list of handles corresponding
17 respectively to objects in the object store.

1 **29.** A set of application program interfaces as recited in claim 26,
2 further comprising:

3 a third interface that enumerates a list of handles corresponding
4 respectively to objects in the object store;

5 wherein the second interface also receives a second handle from the list of
6 handles; and

7 wherein the second interface compares identifying data segments
8 corresponding to the received handles to determine whether the object has been
9 changed in the object store.

10
11 **30.** A set of application program interfaces as recited in claim 26,
12 further comprising a third interface that receives two handles and that returns an
13 indication of whether the two handles correspond to the same object.

14
15 **31.** A set of application program interfaces as recited in claim 26,
16 further comprising:

17 a third interface that receives an object having individual properties and that
18 stores at least some of the individual properties in the object store; and

19 a fourth interface that returns data representing properties of the object that
20 are not supported by the object store.

1 **32.** A set of application program interfaces embodied on a computer-
2 readable medium for execution on a computer in conjunction with an application
3 program that maintains an object store, comprising:

4 a first interface that receives an identifying data segment and that returns a
5 handle corresponding to the identifying data segment;

6 a second interface that enumerates a list of handles corresponding
7 respectively to objects in the object store and to identifying data segments from
8 the respective objects; and

9 a third interface that receives two handles and in response compares the
10 identifying data segments corresponding to the handles and returns an indication
11 of whether the handles represent an unchanged object.

12
13 **33.** A set of application program interfaces as recited in claim 32,
14 further comprising a fourth interface that receives two handles and that returns an
15 indication of whether the two handles correspond to the same object.

16
17 **34.** A set of application program interfaces as recited in claim 32,
18 further comprising:

19 a fourth interface that receives an object having individual properties and
20 that stores at least some of the individual properties in the object store; and

21 a fifth interface that returns data representing properties of the object that
22 are not supported by the object store.

1 **35.** A set of application program interfaces as recited in claim 32,
2 further comprising:

3 a fourth interface that receives two handles and that returns an indication of
4 whether the two handles correspond to the same object;

5 a fifth interface that receives an object having individual properties and that
6 stores at least some of the individual properties in the object store; and

7 a sixth interface that returns data representing properties of the object that
8 are not supported by the object store.

9
10 **36.** A method for synchronizing objects between first and second object
11 stores, wherein the second object store has objects that include properties not
12 supported by the first object store, comprising:

13 sending an object from the second object store to an interface associated
14 with an application program that maintains the first object store;

15 storing at least some individual properties of the object in the first object
16 store;

17 returning data representing unsupported individual properties without
18 storing them in the first object store; and

19 storing the returned data in a reference store that is not implemented by the
20 application program.

21
22 **37.** A method as recited in claim 36, further comprising returning the
23 object, including the unsupported individual properties, to the second object store.

1 **38.** A method as recited in claim 36, further comprising:

2 sending the returned data from the reference store to an interface associated

3 with the application program; and

4 returning the object, including the unsupported individual properties, to the

5 second object store.

6

7 **39.** A system for synchronizing objects between first and second object

8 stores, wherein the second object store has objects that include properties not

9 supported by the first object store, comprising:

10 a primary computer;

11 an application program that executes on the primary computer to maintain

12 the first object store;

13 a synchronization manager that executes on the primary computer;

14 the synchronization manager being configured to send an object from the

15 second store to an interface associated with the application program;

16 the application program and its interfaces being configured to store at least

17 some individual properties of the object in the first object store and to return data

18 representing unsupported individual properties to the synchronization manager

19 without storing them in the first object store; and

20 the synchronization manager being further configured to store the returned

21 data in a reference store.

22

23

24

25

1 **40.** A system as recited in claim 39, wherein:

2 the synchronization manager is further configured to request the object

3 from an interface associated with the application program;

4 wherein the application program and its interfaces receive the data
5 representing the unsupported properties from the synchronization manager and
6 return the object to the synchronization manager; and

7 wherein the synchronization manager sends the object to the second object
8 store.

9

10 **41.** A system as recited in claim 39, further comprising a portable

11 information device that maintains the second object store.

12

13

14

15

16

17

18

19

20

21

22

23

24

25